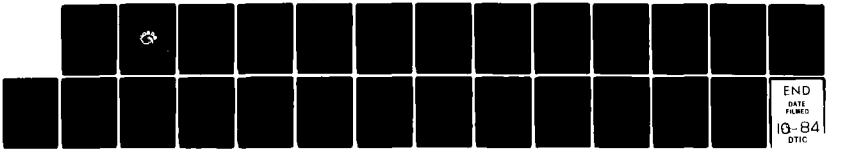
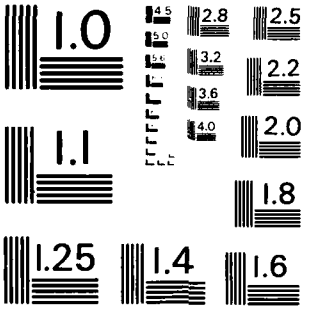
AD-A145 840 CONVERSION OF FEB (FAST AND EASY BINARY) UTILITIES TO ASCII FORTRAN(U) NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL STATION MS W J TEAGUE JUN 84 1/1

UNCLASSIFIED NORDA-TN-275 F/G 9/2 NL



END
DATE FILMED
10-84
DTIC



NORDA Technical Note 275

Naval Ocean Research and
Development Activity
NSTL, Mississippi 39529

Conversion of FEB Utilities to ASCII Fortran

William Teague

Ocean Science Directorate
Oceanography Division

June 1984

ABSTRACT

Software for reading and writing of Fast and Easy Binary (FEB) files in ASCII format is documented in this report. The main FORTRAN-V FEB utilities were rewritten in ASCII FORTRAN and additional programs were developed for conversion of FEB files in FIELDATA format to FEB files in ASCII format, and vice versa. Implementation of these programs will allow FEB file software to continue to be supported by software updates and new additions on the UNIVAC 1180 operating system.

ACKNOWLEDGMENTS

This work was supported by the Physical Oceanography Branch of the Naval Oceanographic Office under Work Request 40043.

TABLE OF CONTENTS

I. INTRODUCTION

During 1975, FEB files were first used at the Rosenstiel School of Marine and Atmospheric Science (University of Miami) for analyzing profiling current meter data. More recently, FEB files have been used at the Naval Ocean Research and Development Activity and at the Naval Oceanographic Office (NOO). Beginning in 1977, software packages were developed at NOO for processing, editing and plotting CTD (conductivity, temperature, depth), XBT (expendable bathythermograph), and current meter data. All programs in these packages are based on FEB files and many are common to all three data types. As new types of data are acquired, programs are written to initially convert them to FEB files, thus making them compatible with existing software. The FEB file structure has played an important role in the data management of physical oceanographic data.

The original FEB programs were entirely written in FORTRAN-V. The NOO UNIVAC no longer supports FORTRAN-V. It is anticipated that by mid 1980s, the ability to compile FORTRAN-V programs on the UNIVAC will be lost, although existing programs should still execute or run. The new industry standard is written in accordance with the specifications of the American National Standards Institute, Inc. (ANSI), in ANSI X3.9-1978 (also known as FORTRAN-77). ASCII FORTRAN is a superset of this standard FORTRAN language. Thus, it has become imperative to rewrite the basic FEB input/output (I/O) routines and utilities in ASCII FORTRAN for operation on the UNIVAC.

The changes made to the basic read and write routines, ZREAD and ZWRIT, are described in Section II. Program listings of the ASCII FORTRAN versions are given in Appendix A. FEB utilities which were converted are described in Section III. Documentation of programs used to convert FEB files in FIELDATA format to FEB files in ASCII format, and vice versa, is contained in Section IV. Program listings of the FIELDATA to ASCII conversion programs are found in Appendix B.

II. CHANGES TO FEB READ AND WRITE ROUTINES

By convention, FEB files are read with the subroutine ZREAD and written with the subroutine ZWRIT. A complete description of these routines is given by Hallock (1980). Changes to these routines were made for ASCII FORTRAN compatibility and for conversion to direct-access READ's and WRITE's.

On the UNIVAC operating system under FORTRAN-V, all character variables are six characters in length. In ASCII FORTRAN, all character-variable lengths must be declared. In keeping with the original number of characters per variable, all character variables were declared CHARACTER*6 in the conversion of ZREAD and ZWRIT.

```
CHARACTER*6 ADOCR,IPR,NMFR,NMBR
```

Within the labeled-common areas, COMMON/RHDR/ was split into two common areas COMMON/RHDR and COMMON/RHDR1. RDHR1 contains character variables previously contained in the old RHDR. The new RHDR contains only the integer variables previously contained in the old RHDR. This change was necessitated by ASCII FORTRAN rules. The new common areas are:

```
COMMON/RHDR/LR,NR,NBR,NFR,NIR,NAR
COMMON/RHDR1/NMBR,NMFR,IPR(1)
```

The original description of FEB files, as given by Hallock (1980), describe sequential, unformatted, non-direct-access files. ZREAD and ZWRIT were updated in 1981 by Teague to reflect a random-access mode. The files were written and read with the same I/O statements; however, the starting locations for the READ's and WRITE's were set with a FORTRAN-V library function SETADR. The address at which to start performing the next I/O operation was calculated from the FEB header COMMON's. Thus, the data records, which were often quite large were skipped over, resulting in faster and more efficient I/O.

In ASCII FORTRAN the ability to randomly access the disk via the call to SETADR is not available. In order to maintain the rapid access of FEB files, a direct-access-file type was chosen where FEB segments are blocked into 600-word records within ZREAD and ZWRIT. The header record is used to calculate the record number for the beginning of the next segment. Data records can then be skipped (emulating SETADR) for fast and efficient I/O. The header information and data are packed such that the header information is always written from the beginning of the 600-word record. To date, all FEB headers have contained less than 600 words. For simplicity, the number of words within the ASCII FEB headers will be restricted to 600 words or less. In equation form the word limitation is given by

$$LR + NFR + NIR + NAR + (3 + LR + NAR)/2 \leq 592$$

where LR is the number of variables, NFR is the number of floating-point numbers (FDOC COMMON), NIR is the number of integer numbers (IDOC COMMON) and NAR is the number of alphanumeric descriptors (ADOC COMMON). The last term on the left side of the equation has arisen because of the change from a six-character FIELDATA word to a four-character ASCII word. Thus the six characters which are contained in one FIELDATA word require one and one-half ASCII words. For example, a FEB file containing 10 variables, 100 IDOC words, and 100 FDOC words, can contain up to 250 ADOC words.

The direct-access ASCII FEB files are opened and file attributes established in ZREAD and ZWRIT through an OPEN statement, which is used to make the connection between the logical unit and the file, as required for read and write operations. The OPEN statement has the form:

```
OPEN(UNIT=IU,ACCESS='DIRECT',FORM='UNFORMATTED',
     STATUS='UNKNOWN',RECL=600,ERR=9090)
```

The IF status flag is set to 1 by the FORTRAN-V ZREAD and ZWRIT upon end-of-file detection prior to RETURN. There is no software end-of-file mark after the last write in a direct-access file. When positioning to the end-of-file with the ASCII ZREAD or ZWRIT, the detection of a read error or finding NBR (segment number) set to zero is used to imply that an end-of-file was encountered, thus conforming to the conventions set by the original ZREAD and ZWRIT.

III. FEB UTILITIES CONVERSION

The following FEB programs were converted from FORTRAN-V to ASCII FORTRAN: ZLOOK, ZDOC, ZMOVE, ZMANGLE, ZSCREEN, ZFILT, ZEDIT, MAINPLOT, and FEBSUM. These non-data-specific utility routines for FEB files have provided a means for listing, restructuring, filtering, editing, plotting, and reviewing FEB files. Basically, the conversion of FEB utilities to ASCII FORTRAN entailed replacing COMMON/RHDR with the

new common areas COMMON/RHDR and COMMON/RHDR1, including the character declaration statement, and making the appropriate call to the ASCII version of ZREAD or ZWRIT. Other areas for conversion included replacing free-field formatted READ's,

```
     READ(5,100) K
100  FORMAT()
```

by list-directed READ's,

```
     READ(5,*) K
```

and splitting common areas containing both character variables and numbers. The documentation of these programs has remained the same and their execution is transparent to the user.

IV. FIELDATA FEB FILE CONVERSION

Large quantities of data are archived in FEB files in the FIELDATA format. As old software continues to be converted to ASCII FORTRAN and new software is written in ASCII FORTRAN, it becomes advantageous to have the FEB data base written in ASCII FORTRAN. This task will become necessary as FORTRAN-V becomes obsolete on the UNIVAC operating system, and will become urgent if the FORTRAN-V compilers are lost. The program FEBCON/ASCII converts the present FIELDATA FEB files to ASCII FEB files. In the event that FORTRAN-V software is required for a FEB file which was written in ASCII FORTRAN, FEBCON/FLDATA converts ASCII FEB files to FIELDATA FEB files. Listings of both of these programs are found in Appendix B.

The program FEBCON/ASCII calls the subroutines ZREAD, ZWRITA, and FFDASC. ZREAD is the FORTRAN-V version and is declared as EXTERNAL in this ASCII FORTRAN program. ZWRITA is very similar to ASCII FORTRAN ZWRIT, with the exception of the ADOC and RHDR common areas. FFDASC is an ASCII FORTRAN library subroutine that allows for conversion from FIELDATA to ASCII. The user input to this program consists of a single line following the prompt:

```
ENTER IU,NS,IS,MSGR,IOU,IOB,MSGW
```

where IU is the logical unit associated with the input FEB file, NS is the number of segments to convert starting at segment number IS, MSGR is the read message level, IOU is the logical unit associated with the output ASCII FEB file, IOB is the segment number of the first segment written in the output file, and MSGW is the write message level.

Similarly, the program FEBCON/FLDATA calls ZREADA, ZWRIT, and FASCFD. ZWRIT is the FORTRAN-V version and is declared as EXTERNAL in this ASCII FORTRAN program. ZREADA is similar to the ASCII FORTRAN ZREAD, with the exception of the ADOC and RHDR common areas. FASCFD is an ASCII FORTRAN library subroutine that allows for conversion from ASCII to FIELDATA. The user input is analogous to FEBCON/ASCII.

V. CONCLUSION

This suite of FEB programs provides a foundation for the transition from FORTRAN-V to ASCII FORTRAN. The philosophy and operation of the programs have remained the same and users should have little difficulty in switching to the ASCII FORTRAN mode of operation. Furthermore, these programs (with the exception of the FIELDATA conversion programs) are easily transportable to other computers, and

presently operate with no modifications on a DIGITAL VAX 11/750. These programs should operate, with few modifications, on the PDP 11/34 shipboard computers.

REFERENCES

Hallock, Z. R., The Fast and Easy Binary (FEB) Data File, Technical Note 7210-12-80, U.S. Naval Oceanographic Office, Bay St. Louis, Ms, 1980.

APPENDIX A

```
      SUBROUTINE ZREAD(IU,IF,IBL)
C
C  THIS SUBROUTINE IS THE READ HALF OF AN INPUT-OUTPUT
C  PACKAGE FOR HANDLING NON-FORMATTED, ASCII FORTRAN
C  WRITTEN DATA FILES, COMMONLY REFERRED TO AS
C  FEB (FAST  EASY BINARY) FILES.
C
C
C
      CHARACTER*6 ADOCR,IPR,NMFR,NMBR
      COMMON /RHDR /LR,NR,NBR,NFR,NIR,NAR
      COMMON /RHDR1 /NMBR,NMFR,IPR(1)
C      NOTE:  ORIGINAL ZREAD CONSISTED OF RHDR COMMON ALONE,
C      ASCII FORTRAN REQUIRED PLACING CHARACTER VARIABLES IN
C      SEPARATE COMMON - RHDR1.
      COMMON /RDOCF/FDOCR(1) /RDOCI/IDOCR(1) /RDOCA/ADOCR(1)
      COMMON / RDATA / VR(1)
C
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
      COMMON / JPOS / JUNIT(30)
      COMMON/DRDCOM/ JFLG,ISECR(30)
      DIMENSION IUNIT(30)
      LOGICAL B1,B210,B10,B35,B45,B69,OD
      DATA MSGR / 2 /
      DATA LLSW/1/, IRST/1/
      DATA IBUF/600/
C
      B1=MSGR.EQ.1
      B210=MSGR.GE.2.AND.MSGR.LE.10
      B10=MSGR.EQ.10
      B35=MSGR.EQ.3.OR.MSGR.EQ.5.OR.MSGR.EQ.7.OR.MSGR.EQ.9.OR.MSGR.EQ.10
      B45=MSGR.EQ.4.OR.MSGR.EQ.5.OR.MSGR.GE.8.AND.MSGR.LE.10
      B69=MSGR.GE.6.AND.MSGR.LE.9
C
C
C      OPEN DIRECT ACCESS FEB FILES AS REQUIRED.
C      RECORD SIZE IS SET TO IBUF IN DATA STATEMENT.
C      EACH HOLLERITH WORD CONSISTS OF SIX CHARACTERS,
C      AND THUS OCCUPIES ONE AND ONE-HALF WORDS.
      IF(IU.NE.IUSAV)THEN
       INQUIRE(UNIT=IU,OPENED=OD)
       IF(.NOT.OD) OPEN(UNIT=IU,ACCESS='DIRECT',FORM='UNFORMATTED',
     *   STATUS='UNKNOWN',ERR=9090,RECL=IBUF)
       IUSAV=IU
      END IF
C
C
      IBLK=IBL
      IPOS=JUNIT(IU)
      IREC=ISECR(IU)
      IF(IPOS.EQ.0) IPOS=1
      IF(IREC.EQ.0)IREC=1
```

```
      IF(IBL.EQ.0) IBLK=IUNIT(IU)
      IF(IBLK.LT.IPOS) GO TO 5
    4 IF(IBLK.EQ.IPOS) GO TO 3
C
C       FULL DUMMY READ IS REQUIRED IN ORDER TO VERIFY RECORD.
C       FILES ARE ZERO-FILLED IN INITIAIZATION ON THE UNIVAC,
C       BUT ARE NOT ON THE VAX - ERR=99 BRANCH IS USED.
       READ(IU'IREC,ERR=99)LQ,NQ,NFQ,NIQ,NAQ,NBQ,(NMBQ,I=1,LQ+2),
     *(FDOCQ,I=1,NFQ),(IDOCQ,I=1,NIQ),(NMBQ,I=1,NAQ),
     *(VQ,M=((IRST-1)*LQ+1),(IRST-1)*LQ+IBUF
     *-(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2))
       IF(LQ.EQ.0.OR.NBQ.EQ.0.OR.NFQ.LT.0.OR.NIQ.LT.0.OR.NAQ.LT.0)
     *GO TO 99
C       IWORDS IS THE TOTAL WORDS CONTAINED IN THE SEGMENT
       IWORDS=(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2)+(LQ*NQ)
C       IREC IS THE RECORD NO. FOR THE NEXT SEGMENT
       IREC=((IWORDS-1)/IBUF)+1+IREC
       IPOS=IPOS+1
       IUNIT(IU)=IPOS
       JUNIT(IU)=IPOS
       GO TO 4
C
    5 IF=0
      IREC=1
      IF(IBL.EQ.0) IBLK=1
      IPOS=1
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      ISECR(IU)=0
      GO TO 4
    3 CONTINUE
C
C     THIS DUMMY READ IS NECESSARY FOR END OF FILE CHECKING,
C     OTHERWISE, ARRAY LIMITS CAN EASILY BE EXCEEDED.
      READ(IU'IREC,ERR=99)LQ,NQ,NFQ,NIQ,NAQ,NBQ,(NMBQ,I=1,LQ+2),
     *(FDOCQ,I=1,NFQ),(IDOCQ,I=1,NIQ),(NMBQ,I=1,NAQ),
     *(VQ,M=((IRST-1)*LQ+1),(IRST-1)*LQ+IBUF
     *-(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2))
      IF(LQ.EQ.0.OR.NBQ.EQ.0.OR.NFQ.LT.0.OR.NIQ.LT.0.OR.NAQ.LT.0)
     *GO TO 99
C
      M1=(IRST-1)*LQ+1
      M2=(IRST-1)*LQ+IBUF-(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2)
C1
      NL=NQ*LQ
      N2=M1+NL-1
      IF(M2.GT.N2)M2=N2
C
C
      READ(IU'IREC,ERR=99)LR,NR,NFR,NIR,NAR,NBR,NMBR,NMFR,
     *(IPR(I),I=1,LR),(FDOCR(J),J=1,NFR),
     *(IDOCR(K),K=1,NIR),(ADOCR(L),L=1,NAR),
     *(VR(M),M=M1,M2)
      IF(NR.GT.NNNR.OR.LR.GT.NNIP.OR.NFR.GT.NNF.
```

```
     *   OR.NIR.GT.NNI.OR.NAR.GT.NNA) GO TO 95
C1
      IF(JFLG.NE.1) GO TO 8
      IWORDS=(8+LR+NFR+NIR+NAR+(3+LR+NAR)/2)+(LR*NR)
C      IREC IS THE RECORD NO. FOR THE LAST RECORD IN THIS SEGMENT
      IREC=((IWORDS-1)/IBUF)+IREC
      GO TO 9
 8      IF(M2.EQ.N2)GO TO 9
      M1=M2+1
      M2=M2+IBUF
      IF(M2.GT.N2)M2=N2
      IREC=IREC+1
      READ(IU'IREC,ERR=99)(VR(J),J=M1,M2)
      GO TO 8
    9 CONTINUE
C
C
      IPOS=IPOS+1
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      ISECR(IU)=IREC+1
C
      IF(MSGR.EQ.0) GO TO 108
      IF(B210) WRITE(6,1000)IU,NMFR,NBR,NMBR,NR,LR,NFR,NIR,NAR
 1000 FORMAT(' READ UNIT',I3,';  FILE ',A6,
     *   ';  SEGNUM',I4,';  SEGNAM ',A6,';  N=',I6,
     *   ';  L=',I4,'  NF=',I4,'  NI=',I4,'  NA=',I4)
C
      IF(B1) WRITE(6,1011) IU,NMFR,NBR,NMBR,NR,LR,NFR,NIR,NAR
 1011 FORMAT(' RD ',I4,2X,A6,2X,I4,2X,A6,2X,I6,4I4)
C
      IF(B35) WRITE(6,1012)(IPR(I),I=1,LR)
 1012 FORMAT(' PARAMETERS: '12(2X,A6)/(13X,12(2X,A6)))
C
      IF(.NOT.B45) GO TO 110
      IF((NFR+NIR+NAR).EQ.0) GO TO 110
      WRITE(6,1013)
 1013 FORMAT(' ADDL DATA:')
      IF(NFR.GT.0)WRITE(6,1100)(FDOCR(I),I=1,NFR)
      IF(NIR.GT.0)WRITE(6,1101)(IDOCR(I),I=1,NIR)
      IF(NAR.GT.0)WRITE(6,1102)(ADOCR(I),I=1,NAR)
 1100 FORMAT(10G11.5)
 1101 FORMAT(1X,12I6)
 1102 FORMAT(1X,12A6)
C
  110 IF(.NOT.B69) GO TO 107
      JL=IRST*LR
      J1=JL-LR+1
      WRITE(6,1014)(VR(I),I=J1,JL)
      JL=(NR+IRST-1)*LR
      J1=JL+1-LR
      WRITE(6,1015)(VR(J),J=J1,JL)
 1014 FORMAT(' FIRST CYCLE:',10G11.5/(13X,10G11.5))
 1015 FORMAT(' LAST CYCLE: ',10G11.5/(13X,10G11.5))
```

```
C
  107 IF(.NOT.B10) GO TO 108
      WRITE(6,1017)
      IQ1=IRST
      IQ2=IQ1+NR-1
      DO 106 I=IQ1,IQ2
      JL=I*LR
      J1=JL+1-LR
      WRITE(6,1016) I,(VR(J),J=J1,JL)
  106 CONTINUE
 1016 FORMAT(5X,I5,3X,10G12.6)
 1017 FORMAT(//' LISTING OF DATA'///)
C
  108 IF=0
      IUP=IU
      RETURN
C
C
   95 IF=5
      WRITE(6,1005)NNNR,NNIP,NNF,NNI,NNA,
     *   NR,LR,NFR,NIR,NAR
 1005 FORMAT(//' A DIMENSION IS TOO SMALL.'//
     *   ' NNNR=',I6,'   NNIP=',I6,'   NNF=',I6,
     *   '   NNI=',I6,'   NNA=',I6//' NR=',I6,
     *   '   LR=',I6,'   NFR=',I6,'   NIR=',I6,'   NAR=',I6//)
      RETURN
   99 IF=1
      WRITE(6,1001) IU
 1001 FORMAT(' EOF ON UNIT ',I3)
 90       IREC=1
      IPOS=0
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      ISECR(IU)=0
      RETURN
 9090      WRITE(6,*) ' ERROR IN OPENING UNIT ',IU
      END
```

```
      SUBROUTINE ZWRIT(JU,IF,IBL)
C
C  THIS SUBROUTINE IS THE WRITE HALF OF AN INPUT-OUTPUT
C  PACKAGE FOR HANDLING NON-FORMATTED, ASCII FORTRAN
C  WRITTEN DATA FILES, COMMONLY REFERRED TO AS
C  FEB (FAST  EASY BINARY) FILES.
C
C
      CHARACTER*6 ADOCR,IPR,NMFR,NMBR
      CHARACTER*6 ADOCW,IPW,NMFW,NMBW
      COMMON / WHDR /LW,NW,NBW,NFW,NIW,NAW
      COMMON / WHDR1 /NMBW,NMFW,IPW(1)
      COMMON /WDOCF/FDOCW(1) /WDOCI/IDOCW(1) /WDOCA/ADOCW(1)
      COMMON / WDATA / VW(1)
C
C      NOTE:  ORIGINAL ZREAD CONSISTED OF RHDR AND WHDR COMMONS ALONE,
C      ASCII FORTRAN REQUIRED PLACING CHARACTER VARIABLES IN
C      SEPARATE COMMONS - RHDR1 AND WHDR1.
C
      COMMON /RHDR /LR,NR,NBR,NFR,NIR,NAR
      COMMON /RHDR1 /NMBR,NMFR,IPR(1)
      COMMON /RDOCF/FDOCR(1) /RDOCI/IDOCR(1) /RDOCA/ADOCR(1)
      COMMON / RDATA / VR(1)
C
C
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
      LOGICAL B1,B210,B10,B35,B45,B69,OD
      COMMON / JPOS / JUNIT(30)
      COMMON /DRDCOM/ JFLG,ISECR(30)
      DIMENSION IUNIT(30)
      DATA MSGW / 2 /
      DATA LLSW / 1 /, IRST, IWST / 1, 1 /
      DATA IBUF/600/
C
      IW=1
      IF(JU.LT.0) IW=2
      IU=ABS(JU)
C
C      OPEN DIRECT ACCESS FEB FILES AS REQUIRED.
C      RECORD SIZE IS SET TO IBUF IN DATA STATEMENT.
C      EACH HOLLERITH WORD CONSISTS OF SIX CHARACTERS,
C      AND THUS OCCUPIES ONE AND ONE-HALF WORDS.
C
      IF(IU.NE.IUSAV)THEN
       INQUIRE(UNIT=IU,OPENED=OD)
       IF(.NOT.OD) OPEN(UNIT=IU,ACCESS='DIRECT',FORM='UNFORMATTED',
     *   STATUS='UNKNOWN',ERR=9090,RECL=IBUF)
       IUSAV=IU
      END IF
C
C
      B1=MSGW.EQ.1
      B210=MSGW.GE.2.AND.MSGW.LE.10
      B10=MSGW.EQ.10
```

```
      B35=MSGW.EQ.3.OR.MSGW.EQ.5.OR.MSGW.EQ.7.OR.MSGW.EQ.9.OR.MSGW.EQ.10
      B45=MSGW.EQ.4.OR.MSGW.EQ.5.OR.MSGW.GE.8.AND.MSGW.LE.10
      B69=MSGW.GE.6.AND.MSGW.LE.9
C
      IBLK=IBL
      IPOS=JUNIT(IU)
      IREC=ISECR(IU)
      IF(IPOS.EQ.0) IPOS=1
      IF(IREC.EQ.0) IREC=1
      IF(IBL.EQ.0) IBLK=IUNIT(IU)
      IF(IBLK.LT.IPOS) GO TO 5
    4 IF(IBLK.EQ.IPOS) GO TO 3
C
C      FULL DUMMY READ IS REQUIRED IN ORDER TO VERIFY RECORD.
C      FILES ARE ZERO-FILLED IN INITIAIZATION ON THE UNIVAC,
C      BUT ARE NOT ON THE VAX - ERR=99 BRANCH IS USED.
      READ(IU'IREC,ERR=99)LQ,NQ,NFQ,NIQ,NAQ,NBQ,(NMBQ,I=1,LQ+2),
     *(FDOCQ,I=1,NFQ),(IDOCQ,I=1,NIQ),(NMBQ,I=1,NAQ),
     *(VQ,M=((IRST-1)*LQ+1),(IRST-1)*LQ+IBUF
     *-(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2))
      IF(LQ.EQ.0.OR.NBQ.EQ.0.OR.NFQ.LT.0.OR.NIQ.LT.0.OR.NAQ.LT.0)
     *GO TO 99
C      IWORDS IS THE TOTAL WORDS CONTAINED IN THE SEGMENT
      IWORDS=(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2)+(LQ*NQ)
C      IREC IS THE RECORD NO. FOR THE NEXT SEGMENT
      IREC=((IWORDS-1)/IBUF)+1+IREC
      IPOS=IPOS+1
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      GO TO 4
C
 5      IREC=1
      IPOS=1
      ISECR(IU)=0
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      IF (IBL.NE.0) GO TO 4
C
C      FIND RECORD NO. FOR WRITE AT END OF FILE, IBL=0
 2      READ(IU'IREC,ERR=6)LQ,NQ,NFQ,NIQ,NAQ,NBQ,(NMBQ,I=1,LQ+2),
     *(FDOCQ,I=1,NFQ),(IDOCQ,I=1,NIQ),(NMBQ,I=1,NAQ),
     *(VQ,M=((IRST-1)*LQ+1),(IRST-1)*LQ+IBUF
     *-(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2))
      IF(LQ.EQ.0.OR.NBQ.EQ.0.OR.NFQ.LT.0.OR.NIQ.LT.0.OR.NAQ.LT.0)
     *GO TO 6
      IWORDS=(8+LQ+NFQ+NIQ+NAQ+(3+LQ+NAQ)/2)+(LQ*NQ)
      IREC=((IWORDS-1)/IBUF)+1+IREC
      IPOS=IPOS+1
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
C2
      GO TO 2
C
    6 CONTINUE
```

```
      WRITE(6,1001)IU
C
C
    3 CONTINUE
      NBW=IPOS
      GO TO (81,82),IW
   81 IF(NW.GT.NNNW.OR.LW.GT.NNIP.OR.NFW.GT.NNF.
     *   OR.NIW.GT.NNI.OR.NAW.GT.NNA) GO TO 95
C
      M1=(IWST-1)*LW+1
      M2=IBUF-(8+LW+NFW+NIW+NAW+(3+LW+NAW)/2)+M1-1
C1
      NL=NW*LW
      N1=(IWST-1)*LW+1
      N2=N1+NL-1
      IF(M2.GT.N2)M2=N2
C
      WRITE(IU'IREC,ERR=97)LW,NW,NFW,NIW,NAW,NBW,NMBW,NMFW,
     *(IPW(I),I=1,LW),(FDOCW(J),J=1,NFW),
     *(IDOCW(K),K=1,NIW),(ADOCW(L),L=1,NAW),
     *(VW(M),M=M1,M2)
C1
       IF(JFLG.NE.1) GO TO 70
        IWORDS=(8+LW+NFW+NIW+NAW+(3+LW+NAW)/2)+(LW*NW)
C       IREC IS THE RECORD NO. FOR THE LAST RECORD IN THIS SEGMENT
        IREC=((IWORDS-1)/IBUF)+IREC
      GO TO 83
 70     IF(M2.EQ.N2)GO TO 83
      M1=M2+1
      M2=M2+IBUF
      IF(M2.GT.N2)M2=N2
      IREC=IREC+1
      WRITE(IU'IREC,ERR=97)(VW(J),J=M1,M2)
      GO TO 70
C
 82     M1=(IRST-1)*LR+1
      M2=IBUF-(8+LR+NFR+NIR+NAR+(3+LR+NAR)/2)+M1-1
C
      NL=NR*LR
      N1=(IRST-1)*LR+1
      N2=N1+NL-1
      IF(M2.GT.N2)M2=N2
C
      WRITE(IU'IREC,ERR=97)LR,NR,NFR,NIR,NAR,NBW,NMBR,NMFR,
     *(IPR(I),I=1,LR),(FDOCR(J),J=1,NFR),
     *(IDOCR(K),K=1,NIR),(ADOCR(L),L=1,NAR),
     *(VR(M),M=M1,M2)
C1
      IF(JFLG.NE.1) GO TO 71
        IWORDS=(8+LR+NFR+NIR+NAR+(3+LR+NAR)/2)+(LR*NR)
C       IREC IS THE RECORD NO. FOR THE LAST RECORD IN THIS SEGMENT
        IREC=((IWORDS-1)/IBUF)+IREC
      GO TO 83
 71     IF(M2.EQ.N2)GO TO 83
```

```
      M1=M2+1
      M2=M2+IBUF
      IF(M2.GT.N2)M2=N2
      IREC=IREC+1
      WRITE(IU'IREC,ERR=97)(VR(J),J=M1,M2)
      GO TO 71
C
   83 CONTINUE
C
C
      IPOS=IPOS+1
      ISECR(IU)=IREC+1
      IUNIT(IU)=IPOS
      JUNIT(IU)=IPOS
      GO TO (84,85),IW
   84 IF(B210) WRITE(6,1000)IU,NMFW,NBW,NMBW,NW,LW,NFW,NIW,NAW
 1000 FORMAT(' WRITE UNIT',I3,';  FILE ',A6,
     *   ';  SEGNUM',I4,';  SEGNAM ',A6,';  N=',I6,
     *   ';  L=',I4,'  NF=',I4,'  NI=',I4,'  NA=',I4)
C
C
      IF(B1) WRITE(6,1011) IU,NMFW,NBW,NMBW,NW,LW,NFW,NIW,NAW
 1011 FORMAT(' WRT ',I4,2X,A6,2X,I4,2X,A6,2X,I6,4I4)
C
      IF(B35) WRITE(6,1012)(IPW(I),I=1,LW)
 1012 FORMAT(' PARAMETERS: '12(2X,A6)/(13X,12(2X,A6)))
C
      IF(.NOT.B45) GO TO 110
      IF((NFW+NIW+NAW).EQ.0) GO TO 110
      WRITE(6,1013)
 1013 FORMAT(' ADDL DATA:')
      IF(NFW.GT.0)WRITE(6,1100)(FDOCW(I),I=1,NFW)
      IF(NIW.GT.0)WRITE(6,1101)(IDOCW(I),I=1,NIW)
      IF(NAW.GT.0)WRITE(6,1102)(ADOCW(I),I=1,NAW)
 1100 FORMAT(10G11.5)
 1101 FORMAT(1X,12I6)
 1102 FORMAT(1X,12A6)
C
  110 IF(.NOT.B69) GO TO 107
      JL=IWST*LW
      J1=JL-LW+1
      WRITE(6,1014)(VW(I),I=J1,JL)
      JL=(NW+IWST-1)*LW
      J1=JL+1-LW
      WRITE(6,1015)(VW(J),J=J1,JL)
 1014 FORMAT(' FIRST CYCLE:',10G11.5/(13X,10G11.5))
 1015 FORMAT(' LAST CYCLE: ',10G11.5/(13X,10G11.5))
C
  107 IF(.NOT.B10) GO TO 108
      WRITE(6,1017)
      IQ1=IWST
      IQ2=IQ1+NW-1
      DO 106 I=IQ1,IQ2
      JL=I*LW
```

```
      J1=JL+1-LW
      WRITE(6,1016) I,(VW(J),J=J1,JL)
  106 CONTINUE
 1016 FORMAT(5X,I5,3X,10G12.6)
 1017 FORMAT(//' LISTING OF DATA'///)
C
      GO TO 86
   85 IF(B210) WRITE(6,1000)IU,NMFR,NBW,NMBR,NR,LR,NFR,NIR,NAR
C
      IF(B1) WRITE(6,1011) IU,NMFR,NBW,NMBR,NR,LR,NFR,NIR,NAR
C
      IF(B35) WRITE(6,1012)(IPR(I),I=1,LR)
C
      IF(.NOT.B45) GO TO 109
      IF((NFR+NIR+NAR).EQ.0) GO TO 109
      WRITE(6,1013)
      IF(NFR.GT.0)WRITE(6,1100)(FDOCR(I),I=1,NFR)
      IF(NIR.GT.0)WRITE(6,1101)(IDOCR(I),I=1,NIR)
      IF(NAR.GT.0)WRITE(6,1102)(ADOCR(I),I=1,NAR)
C
  109 IF(.NOT.B69) GO TO 117
      JL=IRST*LR
      J1=JL+1-LR
      WRITE(6,1014)(VR(I),I=J1,JL)
      JL=(NR+IRST-1)*LR
      J1=JL+1-LR
      WRITE(6,1015)(VR(J),J=J1,JL)
C
  117 IF(.NOT.B10) GO TO 108
      WRITE(6,1017)
      IQ1=IRST
      IQ2=IQ1+NR-1
      DO 116 I=IQ1,IQ2
      JL=I*LR
      J1=JL+1-LR
      WRITE(6,1016) I,(VR(J),J=J1,JL)
  116 CONTINUE
C
C
   86 CONTINUE
  108 IF=0
      IUP=IU
      RETURN
C
C
   95 IF=5
      WRITE(6,1005)NNNW,NNIP,NNF,NNI,NNA,
     *   NW,LW,NFW,NIW,NAW
 1005 FORMAT(//' A DIMENSION IS TOO SMALL.'//
     *   ' NNNW=',I6,'   NNIP=',I6,'   NNF=',I6,
     *   '   NNI=',I6,'   NNA=',I6//' NW=',I6,
     *   '   LW=',I6,'   NFW=',I6,'   NIW=',I6,'   NAW=',I6//)
      RETURN
   97 IF=3
```

```
      WRITE(6,1003) IU
 1003 FORMAT(' WRITE ERROR ON UNIT ',I3)
      LU=IU
      GO TO 90
   99 IF=1
      WRITE(6,1001) IU
 1001 FORMAT(' EOF ON UNIT ',I3)
      LU=IU
      GO TO 90
C
      ENTRY RESETW(KU)
      LU=KU
      CLOSE(UNIT=LU)
      IUSAV=0
 90      IREC=1
      IPOS=0
      IUNIT(LU)=0
      JUNIT(LU)=0
      ISECR(LU)=0
      RETURN
 9090      WRITE(6,*) ' ERROR IN OPENING UNIT ',IU
      END
```

APPENDIX B

```
C*********************************************************************
C
C       PROGRAM: FEBCON/ASCII
C       PURPOSE: THIS PROGRAM READS A FEB FILE WRITTEN IN
C                FIELD DATA AND REWRITES THE FEB FILE IN
C                ASCII CHARACTERS.
C
C*********************************************************************
      CHARACTER*6 NMBR1,NMFR1,IPR1,ADOCR1
      COMMON / RHDR / LR,NR,NBR,NMBR,NMFR,NFR,NIR,NAR,IPR(20)
      COMMON /RHDR2 /LR2,NR2,NBR2,NFR2,NIR2,NAR2
      COMMON /RHDR1 /NMBR1,NMFR1,IPR1(20)
      COMMON /RDOCF/FDOCR(125) /RDOCI/IDOCR(125) /RDOCA/ADOCR(200)
       COMMON /RDOCA1/ADOCR1(200)
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
      COMMON / RDATA / VR(24000)
      EXTERNAL ZREAD(FOR)
      DATA NNIP,NNI,NNA/20,125,200/,NNF/125/
C
      NNNR=6000
 101      WRITE(6,*) 'ENTER IU,NS,IS,MSGR,IOU,IOB,MSGW'
      READ(5,*,=99)IU,NB,IB,MSGR,IOU,IOB,MSGW
      WRITE(6,1003)IU,IOU,NB,IB,IOB
 1003 FORMAT(' INPUT UNIT=',I4,'   OUTPUT UNIT=',I4,
     *   '   NO. OF SEGS=',I6,'   START SEG/R=',I6,
     *   '   START SEG/W=',I6/)
      IF(IOB.LE.0)IOB=0
      IOU1=-IOU
      DO 4 I=1,NB
      CALL ZREAD(IU,IF,IB)
      IF(IF.NE.0) GO TO 98
      IB=0
      LR2=LR
      NR2=NR
      NBR2=NBR
      NFR2=NFR
      NIR2=NIR
      NAR2=NAR
      IWC=NAR
C      SUBROUTINE FFDASC IS FOUND IN THE ASCII FORTRAN LIBRARY.
      CALL FFDASC(IWC,ADOCR,ADOCR1)
      IWC=1
      CALL FFDASC(IWC,NMBR,NMBR1)
      IWC=1
      CALL FFDASC(IWC,NMFR,NMFR1)
      IWC=LR
      CALL FFDASC(IWC,IPR,IPR1)
      CALL ZWRITA(IOU1,IF,IOB)
      IOB=0
    4 CONTINUE
      GO TO 101
C
```

```
   98 IF(IF.GT.1) GO TO 100
      WRITE(6,9800)
 9800 FORMAT(/' EOF ON INPUT. NEXT      '/)
      GO TO 101
   99 WRITE(6,6600)
 6600 FORMAT(//'  OF JOB')
      STOP
C
   97 WRITE(6,9700) NB,IB
 9700 FORMAT(//' **  INPUT ERROR.  NO. OF SEGS=',I5,'  START SEG/R=',I5)
      STOP
C
 100      CONTINUE
      WRITE(6,8000)IF
 8000      FORMAT('  IF=',I6)
      STOP
```

```
C*********************************************************************
C
C      PROGRAM: FEBCON/FLDDATA
C      PURPOSE: THIS PROGRAM READS A FEB FILE WRITTEN IN
C               ASCII AND REWRITES THE FEB FILE IN
C               FIELD DATA.
C
C*********************************************************************
      CHARACTER*6 NMBR1,NMFR1,IPR1,ADOCR1
      COMMON / RHDR / LR,NR,NBR,NMBR,NMFR,NFR,NIR,NAR,IPR(20)
      COMMON /RHDR2 /LR2,NR2,NBR2,NFR2,NIR2,NAR2
      COMMON /RHDR1 /NMBR1,NMFR1,IPR1(20)
      COMMON /RDOCF/FDOCR(125) /RDOCI/IDOCR(125) /RDOCA/ADOCR(200)
       COMMON /RDOCA1/ADOCR1(200)
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
      COMMON / RDATA / VR(24000)
       DIMENSION NMFR2(2)
      EXTERNAL ZWRIT(FOR)
      DATA NNIP,NNI,NNA/20,125,200/,NNF/125/
C
      NNNR=6000
 101      WRITE(6,*) 'ENTER IU,NS,IS,MSGR,IOU,IOB,MSGW'
      READ(5,*,END=99)IU,NB,IB,MSGR,IOU,IOB,MSGW
      WRITE(6,1003)IU,IOU,NB,IB,IOB
 1003 FORMAT(' INPUT UNIT=',I4,'   OUTPUT UNIT=',I4,
     *   '   NO. OF SEGS=',I6,'   START SEG/R=',I6,
     *   '   START SEG/W=',I6/)
      IF(IOB.LE.0)IOB=0
      IOU1=-IOU
      DO 4 I=1,NB
      CALL ZREADA(IU,IF,IB)
      IF(IF.NE.0) GO TO 98
      IB=0
      LR=LR2
      NR=NR2
      NBR=NBR2
      NFR=NFR2
C      WRITE(6,*)'NFR=',NFR
      NIR=NIR2
      NAR=NAR2
      IWC=NAR2+((NAR2+1)/2)
C      SUBROUTINE FASCFD IS FOUND IN THE ASCII FORTRAN LIBRARY.
      CALL FASCFD(IWC,ADOCR1,ADOCR)
      IWC=2
      CALL FASCFD(IWC,NMBR1,NMBR)
      IWC=2
      CALL FASCFD(IWC,NMFR1,NMFR2)
      NMFR=NMFR2(1)
C      WRITE(6,*)'NFR=',NFR
      IWC=LR2+((LR2+1)/2)
      CALL FASCFD(IWC,IPR1,IPR)
C      WRITE(6,*)'LR,NR,NBR,NFR,NAR,NIR',LR,NR,NBR,NFR,NAR,NIR
      CALL ZWRIT(IOU1,IF,IOB)
      IOB=0
```

```
    4 CONTINUE
      GO TO 101
C
   98 IF(IF.GT.1) GO TO 100
      WRITE(6,9800)
 9800 FORMAT(/' EOF ON INPUT. NEXT      '/)
      GO TO 101
   99 WRITE(6,6600)
 6600 FORMAT(//' END OF JOB')
      STOP
C
   97 WRITE(6,9700) NB,IB
 9700 FORMAT(//' **  INPUT ERROR.  NO. OF SEGS=',I5,'  START SEG/R=',I5)
      STOP
C
 100      CONTINUE
      WRITE(6,8000)IF
 8000      FORMAT('  IF=',I6)
      STOP
      END
```

```
      SUBROUTINE ZREADA(IU,IF,IBL)
      CHARACTER*6 ADOCR,IPR,NMFR,NMBR,NMBQ
      COMMON /RHDR2 /LR,NR,NBR,NFR,NIR,NAR
      COMMON /RHDR1 /NMBR,NMFR,IPR(1)
      COMMON /RDOCF/FDOCR(1) /RDOCI/IDOCR(1) /RDOCA1/ADOCR(1)
      COMMON / RDATA / VR(1)
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
C     COMMON / JPOS / JUNIT(30)
C     COMMON/DRDCOM/ JFLG,ISECR(30)
                        .
                        .
                        .
                        .
                        .
            Same as ASCII FORTRAN ZREAD
                        .
                        .
                        .
                        .
                        .

      SUBROUTINE ZWRITA(JU,IF,IBL)
      CHARACTER*6 ADOCR,IPR,NMFR,NMBR,NMBQ
      CHARACTER*6 ADOCW,IPW,NMFW,NMBW
      COMMON / WHDR /LW,NW,NBW,NFW,NIW,NAW
      COMMON / WHDR1 /NMBW,NMFW,IPW(1)
      COMMON /WDOCF/FDOCW(1) /WDOCI/IDOCW(1) /WDOCA/ADOCW(1)
      COMMON / WDATA / VW(1)
      COMMON /RHDR2 /LR,NR,NBR,NFR,NIR,NAR
      COMMON /RHDR1 /NMBR,NMFR,IPR(1)
      COMMON /RDOCF/FDOCR(1) /RDOCI/IDOCR(1) /RDOCA1/ADOCR(1)
      COMMON / RDATA / VR(1)
      COMMON / DIAGS / MSGR,MSGW,NNNR,NNNW,NNIP,NNF,NNI,NNA,IRST,IWST
      LOGICAL B1,B210,B10,B35,B45,B69,OD
C     COMMON / JPOS / JUNIT(30)
C     COMMON /DRDCOM/ JFLG,ISECR(30)
                        .
                        .
                        .
                        .
                        .
            Same as ASCII FORTRAN ZWRIT
                        .
                        .
                        .
                        .
                        .
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Technical Note 275	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Conversion of FEB utilities to ASCII FORTRAN		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* William J. Teague		8. CONTRACT OR GRANT NUMBER*(s)* N6230684WR40043
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Research & Development Activity NSTL, MS 39529		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Ocean Research & Development Activity NSTL, MS 39529		12. REPORT DATE ~~June~~ May 1984 13. NUMBER OF PAGES 24
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Distribution Unlimited

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

FEB

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Software for reading and writing of Fast and Easy Binary (FEB) files in ASCII format is documented in this report. The main FORTRAN-V FEB utilities were rewritten in ASCII FORTRAN and additional programs were developed for conversion of FEB files in FIELDATA format to FEB files in ASCII format, and vice versa. Implementation of these programs will allow FEB file software to continue to be supported by software updates and new additions on the UNIVAC 1108 operating system.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*


DATE
ILME